A Lightweight Aggregate Authentication Protocol for Internet of Drones

Image Bhattarai[‡] Cong Pu[‡] Kim-Kwang Raymond Choo[¶]
[‡]Oklahoma State University, United States. Email: image.bhattarai@okstate.edu; cong.pu@outlook.com
[¶]The University of Texas at San Antonio, United States. Email: raymond.choo@fulbrightmail.org

***Abstract*—The Internet of Drones (IoD), an innovative aerial-ground communication architecture, has quickly became the driving force for various civilian applications (e.g., body temperature detecting drones during the global pandemic of coronavirus disease). In the IoD, a fleet of drones are deployed over an area of interest, collect task-specific data, and then deliver them to the ground station for further data exploration and analysis. To fully exploit the potential of IoD in today's dynamic and evolving cyber-threat environment, the security and efficiency challenges existing in the IoD communications should be well addressed. Some researchers have developed security mechanisms to enable the authentication between the ground station and the drones in the IoD systems. Nonetheless, those schemes mainly focus on the security aspect but overlook the importance of communication efficiency to the resource-constrained drones. In order to fill this research gap, this paper proposes a lightweight aggregate authentication scheme (hereafter referred to as *liteAGAP*) to tackle the challenges of communication security and efficiency together. Specifically, *liteAGAP* utilizes cryptographic primitives such as physical unclonable function and bilinear pairing to efficiently secure the data exchange between the ground station and a group of drones in the IoD systems. To evaluate its security performance, *liteAGAP* is first implemented in the security-sensitive protocol modeling language. Then, we analyze and verify *liteAGAP* using AVISPA, which is a well-known Internet security protocol verification framework. We also implement *liteAGAP* and its counterpart schemes in a simulation environment, where the simulation-based experiments are conducted to obtain the results of communication overhead, running time, memory storage usage, and energy consumption. According to the results of security verification/analysis and performance evaluation, we conclude that not only *liteAGAP* meets the expected security requirements, but also provides superior performance compared to the existing schemes.**

***Index Terms*—Internet of Drones, Security, Aggregate Authentication, Physical Unclonable Function, Bilinear Pairing**

I. Introduction

The Internet of Things (IoT) applications normally consist of a set of immobile sensors, which are connected to the back-end data collection server via wired/wireless communication systems [1]. In recent years, drones have begun to efficiently replace connected sensors ''at rest'' with one device that is moveable within different environments, adequate to equip various sensors/devices, adaptable to diverse tasks, and intelligent to collect data on anything, anytime, and anywhere [2]. Inspired by the idea of IoT, there has been a constant effort to keep the momentum forward on the ubiquitous computing and bring forth an innovative aerial-ground communication architecture, which is termed the Internet of Drones (IoD) [3]. As drones are being integrated with other technologies (e.g., artificial intelligence), we will see more IoD systems/applications performing critical missions/tasks, especially where it is costly, risky or impractical for humans to perform [4]. In these scenarios, drones are able to complete missions/tasks in a more efficient and less risky manner [5]. In comparison with vehicular networks [6], where the road infrastructure restricts the movement of vehicles, the IoD drones are provided with more movement flexibility while executing missions/tasks in various areas of interest. Additionally, with the assistance of drones, a considerable amount of manpower can be released and the road traffic can be shifted to the airspace (i.e., thermal imaging and disinfecting drones for COVID-19 [7]), resulting in the improvement of transportation congestion and safety.

As a new generation of mobile computing network, one advantageous feature of IoD is the bidirectional communication, where the real-time command/instruction and information/data are seamlessly exchanged between the ground station and the drones [8]. To realize the mission/task objectives, a group of drones are deployed over an area of interest, collect the relevant data, and then periodically report them to the ground station for further data exploration and analysis. Nevertheless, in today's evolving and dynamic cyber-threat environment, data communication should be not only secure but also efficient to the resource-limited drones [9]. First, as the data communication is through wide-open wireless medium, the adversary can sniff, or even further spoof and transmit the contaminated data to the ground station. As a result, the process of data exploration and analysis will fail and the wrong command/instruction could be made by the ground station (i.e., sending a drone to the wrong location for data collection). Second, if a group of drones communicate with the ground station for data exchange simultaneously, a severe signaling congestion will occur at the ground station. Obviously, drones might be faced with data transmission failure or denial of service, and the overall quality of service will be adversely affected (i.e., the authentication process fails).

Lately, several data aggregation techniques have been investigated to reduce data redundancy and improve communication efficiency in the IoT setting [10]–[12]. Those techniques are mainly designed for stationary IoT devices, however, they cannot be exploited in the IoD environment because of intermittent network connectivity between drones and static network structure. Additionally, a few studies on mutual authentication have been conducted in the IoD setting [13]–[15], where the drones and the ground station exchange pre-

979-8-3503-0457-2/24/$31.00 ©2024 IEEE

synchronized secrets so that they can verify each other's credibility and feel confident to set up a secret key for further communication. Unfortunately, the existing security schemes only concentrate on the security aspect of IoD communications but completely overlook the importance of communication efficiency to drones with limited resources. In order to get the full benefits out of IoD paradigm, both security and efficiency issues of IoD communication should be addressed concurrently in the design of security protocol.

In order to bridge the above-mentioned research gap, this paper proposes a lightweight aggregate authentication scheme, hereafter referred to as *liteAGAP*, to provide secure and efficient IoD communications. In the *liteAGAP*, a group of drones gather data about specific subjects and transmit them along with their digital signatures to the aggregation drone. After receiving the data and digital signatures from other drones, the aggregation drone aggregates all drones' data and digital signatures, and delivers them to the ground station. To evaluate the security and performance of *liteAGAP*, we first validate the security properties of *liteAGAP* using AVISPA [16] which is a well-known Internet security protocol verification framework. Then, we develop an experimental testbed, implement *liteAGAP* and two benchmark schemes (PPAAS [17] and GASE [18]), and conduct comprehensive simulation-based experiments. Our experimental study indicates that *liteAGAP* not only meets the expected security requirements but also achieves superior performances while comparing with its counterparts.

The novelty of our work can be justified from two different perspectives: Internet of Drones (IoD) and the integration of security and efficiency. First, the IoD paradigm has become an active field of research in the recent past, and is of great interest to many technical communities and commercial companies, e.g., IEEE Communications Society [19], Ericsson [20], etc. The research outcomes of this paper will provide a thorough understanding of IoD architecture as well as its unique security and performance challenges and requirements. Second, our approach not only focuses on the security aspect of IoD data communication, but also attempts to boost the efficiency of data communication between the drones and the ground station. There are existing security mechanisms in the IoD environment, however, they fail to take into consideration the efficiency of data communication, but only protect the IoD data communication.

II. Related Work

In [21], the authors develop a security solution for drone swarm communication. Initially, the ground station registers all drones through storing their physical unclonable function (PUF) challenge–response pairs (CRP). Before exchanging any critical information, the ground station and the drones need to go through the process of authentication and session key establishment. The ground station first sends an authentication request message to the drone swarm network, where the authentication request message will be delivered to every drone through hop-by-hop cooperative relay. Then, each drone replies the authentication response message which is forwarded through intermediate drones and finally reaches the ground station. During this process, the intermediate drone also attaches its response in the received authentication response message from other drones. After receiving the aggregated response message, the ground station creates session keys and sends them to each drone in the drone swarm network. One obvious drawback of this security scheme is that frequently receiving and forwarding authentication request and response messages consume drones' limited energy resource. In addition, the authentication procedure will fail if either the authentication request or response message gets lost during the transmission due to bad channel quality.

The authors in [22] propose an elliptic-curve cryptosystem (ECC) based security solution so that the user and its associated drone can communicate securely. During the initial phase, the ground station initializes system parameters and registers the user and its drone by exchanging mutual-agreed information (i.e., pseudonym, password, and biometrics). After that, the legitimate user utilizes its password and biometrics to authenticate with its associated drone through the ground station, and establishes a session key for the exchange of critical information by using ECC. Another security feature is that the user can change its password and biometrics to defend against brute force attack. One shortcoming of this scheme is that it just performs authentication between one user and one drone, and does not support many-to-one authentication (e.g., authentication between a group of drones and one user).

In [23], the authors adopt federated learning technique to train deep neural network model with the radio frequency of drones and achieve mutual authentication between the ground station and the drones. The advantage of using federated learning is that it is unnecessary to synchronize the system setting between drones and the ground station. The authors [24] design a group authentication protocol for drone networks, where the new drone is verified by the group leading drone before it can join the drone network and communicate with other drones. In [25], a delegation based authentication scheme is proposed for device-to-device networks, where the drone uses its proxy signature to authenticate itself with other drones in the network. The authors in [26] develop a handover authentication mechanism so that the performance of handover process can be improved when the vehicular platoon changes the contact point of aerial networks in space-air-ground integrated vehicular networks. However, all the abovementioned studies fail to suggest the security solution through which a group of drones and the ground station can securely and efficiently exchange IoD data simultaneously.

A blockchain-enabled authentication scheme is proposed for industrial drone networks [27], where the blockchain network is responsible for storing drone's authentication information. To achieve authentication between the user and the ground station, they can request to retrieve and/or update the authentication information stored in the blockchain network. The authors also suggest an approach to reach an agreement on a shared session key (or called group key) among a group of drones. However, the establishment of a group key

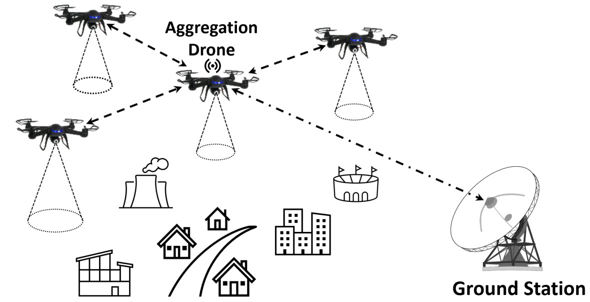


Fig. 1. Network model.

between the ground station and a group of drones is still missing. In [28], a data type sensitive authentication protocol is proposed for IoD environment, where the ground station will assist with the authentication between the user and the drone. The shining point of the proposed security scheme is that the data collected by drones can be distinguished for various applications in terms of data type. Unfortunately, this approach is only designed for one-to-one communication (i.e., the communication between the user and the drone), and does not support either group authentication or data aggregation.

In short, many researchers spent effort on the security issues of IoD systems and developed various security mechanisms. However, they are not giving much attention to lightweight aggregate authentication protocols to address the security and efficiency of IoD communications simultaneously.

III. Preliminary Background

A. Network Model and Security Requirements

As shown in Fig. 1, a group of drones are deployed over an area of interest, collect task-related data, and send them to the ground station. This scenario has a wide range of applications, e.g., a drone swarm surveils a crowd of street demonstrators and delivers surveillance data to the ground station for estimating the size of demonstration and its movement [29]. Since the data/command exchange between the drones and the ground station is carried out over an open and vulnerable wireless medium, the malicious attackers have means to interfere with their interactions, according to Dolev–Yao adversarial model [30]. As a result, the drones and the ground station are required to validate each other's identifications before performing any data/command exchange. When a number of drones are about to set up secure communication channels with the ground station for data/command exchange, they might send authentication request messages to the ground station simultaneously, causing the problem of authentication signaling congestion at the ground station. To establish secure and efficient communications between the drone swarm and the ground station, one drone is elected to serve as the aggregation drone who accepts the responsibility for gathering the data from other drones, and subsequently aggregating and sending them to the ground station.

The main security objectives of *liteAGAP* is to provide entity authentication guarantee and data integrity service, as well as improve communication efficiency. First, *liteAGAP* shall facilitate the identity verification among a group of drones, the aggregation drone, and the ground station, and provide confidence in the secure exchange of information. Second, *liteAGAP* shall guarantee that the message origin or content can be validated by the recipient of message. Third, *liteAGAP* shall lessen the communication and computation overhead when a drone swarm authenticates with the ground station for data exchange.

TABLE I
Notations

Notation	Meaning
GS_k	The ground station GS_k
n_i	The drone n_i
n_a	The aggregation drone n_a
che_i	Drone n_i's PUF challenge
res_i	Drone n_i's PUF response
$(che_i,\ res_i)$	Drone n_i's challenge-response pair (CRP)
$F^i_{puf}(\cdot)$	Drone n_i's PUF
$rGen(\cdot)$	PUF response generation algorithm
$rRes(\cdot)$	PUF response restore algorithm
S	Helper string
m	Modulus m
$H_a(\cdot)$	Hash function, H_a:{0,1}*→$\mathbb{G}$
$H_b(\cdot)$	Hash function, H_b:{0,1}*→$\mathbb{Z}$
$\|$	Concatenation operator
s	Private key of GS_k
T	Public key of GS_k
T_i	Public key of n_i
x	Random number
DS	Digital signature of GS_k
ts	Timestamp
d_i	n_i's collected data
DS_i	Digital signature of n_i
DS^*	Aggregated Digital signature
D	All collected data
$\mathbb{G}$	Cyclic additive group
P	Arbitrary generator of $\mathbb{G}$
q	The order of $\mathbb{G}$
$\mathbb{G}_T$	Cyclic multiplicative group
n	The order of $\mathbb{G}$ and $\mathbb{G}_T$
$\hat{e}$: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$	Bilinear pairing map on ($\mathbb{G}$, $\mathbb{G}_T$)

B. Bilinear Pairing

$\mathbb{G} = \langle P \rangle$ is defined as a cyclic additive group, and q and P indicate the order of $\mathbb{G}$ and an arbitrary generator of $\mathbb{G}$, respectively. In addition, a multiplicative cyclic group, which is denoted as $\mathbb{G}_T$, is created with an identical order q. Here, an extremely large prime number (i.e., at least 1024-bit) is usually assigned to q. In summary, a bilinear pairing map on ($\mathbb{G}$, $\mathbb{G}_T$) designed as $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is born with the following characteristics [31]:

1) $\hat{e}(P + Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$ and $\hat{e}(P, Q + R) = \hat{e}(P, Q)\hat{e}(P, R)$, where P, Q, and $R \in \mathbb{G}$.
2) $\hat{e}(\alpha P, \beta Q) = \hat{e}(P, Q)^{\alpha\beta}$, where α and $\beta \in \mathbb{Z}$.
3) $\hat{e}(P, P) \neq 1$. Here 1 is the identity element of $\mathbb{G}_T$.
4) $\hat{e}(P, Q) = \hat{e}(Q, P)$.
5) $\hat{e}$ can be efficiently calculated.

The key idea of bilinear pairing map relies on the intractability of computational Diffie-Hellman problem, where it is highly impossible to calculate $\alpha\beta P \in \mathbb{G}$ (or calculate

```
Algorithm 1: Response Generation Algorithm rGen
  Input: Modulus $m$; Challenge $che$
Function rGen($m$, $che$):
    /* $\xleftarrow{\circledast}$ denotes sampling */
    /* $\oplus$ denotes exclusive OR function */
    /* $\mathbb{Z}_m$ denotes the set of remainders in
       arithmetic modulo m */
    $O = F_{puf}(che)$;
    $res \xleftarrow{\circledast} \mathbb{Z}_m$;
    $S = O \oplus ECC(res)$;
    return $\{res, S\}$;
```

```
Algorithm 2: Response Restore Algorithm rRes
  Input: Challenge $che$; Helper string $S$
Function rRes($che$, $S$):
    $O^{'} = F_{puf}(che)$;
    $res = D_{er}(S \oplus O^{'})$;
    return $res$;
```

$\hat{e}(P,\, P)^{\alpha\beta\gamma})$ within polynomial time, given P, αP, βP, and γP. Nonetheless, it is effortless to verify whether $\gamma P = \alpha\beta P$ (or $\alpha\beta = \gamma \bmod n$) by checking $\hat{e}(\alpha P,\, \beta P) \overset{?}{=} \hat{e}(P,\, \gamma P)$, which is widely known as the decisional Diffie-Hellman problem.

C. Physical Unclonable Function

Physical unclonable functions, also widely known as PUFs, take advantage of the unique physical irregularity of integrated circuit to realize one-to-one mapping between input query and specific output [32]. In this context, the input query has a specific name, called PUF challenge. Correspondingly, the specific output is called PUF response. The PUF challenge che combined with the corresponding response res are called challenge-response pair, or PUF CRP for short. In general, the PUF can be represented or simulated as a secure one-way function F_{puf}, where $res = F_{puf}(che)$. An interesting feature of PUF is that the PUF always produces the same response res when it is fed with the identical challenge che every time. However, if the different challenges are provided to the PUF, the completely distinct responses can be expected.

In noisy environments, the identical challenges fed to the PUF might not be able to get the same responses [33]. In other words, the PUF is sensitive to external environment changes/noise, thus, the secret data of cryptographic operations might not be regenerated by the PUF. To resolve this important issue, error correction code (ECC) and fuzzy extractor can be integrated with the PUF. First, we define an algorithm *rGen* to generate the response. The *rGen* algorithm will output a set $\{res, S\}$. Here, *res* is the CRP response, which is the value to be regenerated by the PUF. *S* is a helper string which is fed into the PUF to regenerate the CRP response *res*. The error correction code (ECC) [34] is adopted to eliminate up to *x* bit errors in the CRP response *res*.

We also design a response restore algorithm, denoted as $rRes$. The main purpose of $rRes$ is to allow the PUF to restore the CRP response *res* with the assistance of the helper string *S* and the error decoding algorithm D_{er}, even if the PUF produces an output $O^{'}$ that differs from the original output O by at most *x* bits.

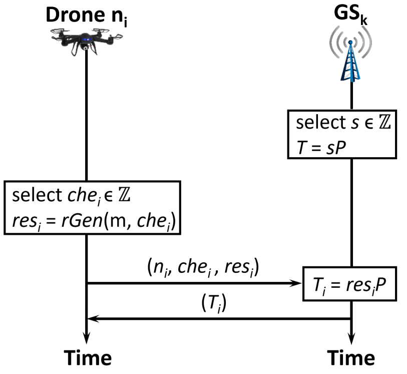


Fig. 2. System setup phase.

IV. The Proposed *liteAGAP* Protocol

liteAGAP consists of three phases: (i) system setup; (ii) data request; and (iii) data response. Without loss of generality, we assume that a group of *j* drones (denoted by $N = \{n_1,\, n_2,\, \cdots,\, n_j\}$) and a ground station GS_k are deployed in the area of operations.

A. System Setup Phase

First, the ground station GS_k chooses two groups ($\mathbb{G}$ and $\mathbb{G}_T$) of the same prime order q, where $\mathbb{G}$ and $\mathbb{G}_T$ are the cyclic additive group and the multiplicative group, respectively. Let P be an arbitrary generator of $\mathbb{G}$. GS_k generates a bilinear pairing map $\hat{e}$ on $\mathbb{G}$ and $\mathbb{G}_T$, $\hat{e}$: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. Then, GS_k chooses two cryptographic hash functions H_a and H_b. Here, H_a : $\{0{,}1\}^{*} \rightarrow \mathbb{G}$ and H_b : $\{0{,}1\}^{*} \rightarrow \mathbb{Z}$.

Second, GS_k randomly selects a number $s \in \mathbb{Z}$ as its private key, and calculates the corresponding public key $T = sP$. Moreover, each drone $n_i \in N$ selects the challenge $che_i \in \mathbb{Z}$ and computes the corresponding response res_i as its private key. Then, n_i registers at GS_k by submitting its res_i securely. GS_k uses n_i's res_i to calculate its public key $T_i = res_iP$, and sends T_i back to n_i. When the above process is complete, n_i only stores its che_i and T_i, while GS_k keeps n_i's che_i, res_i, and T_i. Note that n_i does not directly store res_i in the memory to defend against drone capture and power analysis attacks. Finally, GS_k publishes the following unclassified system parameters, $\{\mathbb{G},\, \mathbb{G}_T,\, r,\, P,\, H_a,\, H_b,\, T_i\}$. Fig. 2 presents the system setup process of *liteAGAP*.

B. Data Request Phase

To collect task-related data, the ground station GS_k regularly broadcasts a data request message to all drones in the area of operations. First, GS_k randomly selects a number $x \in \mathbb{Z}$ and calculates $R = xP$. Then, GS_k creates its digital signature $DS = sH_a(GS_k||ts||R)$, where s is its private key and ts is the current system time. Finally, GS_k generates a data request message piggybacked with its digital signature, which is represented as *dREQ* = $\{GS_k,\, ts,\, R,\, DS\}$, and broadcasts it to all drones in the area.

After receiving the data request message *dREQ*, each drone $n_i \in N$ first verifies the piggybacked timestamp ts of *dREQ*

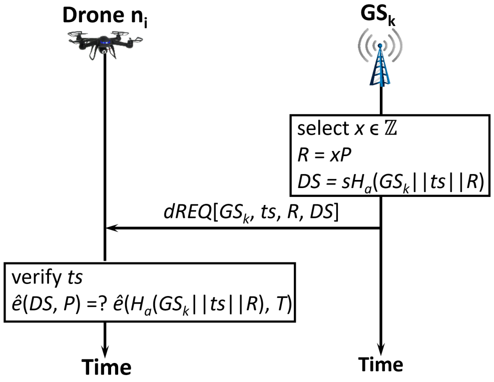


Fig. 3. Data request phase.

for the potential replay attack. If ts is obsolete, n_i will discard *dREQ* directly because *dREQ* might be a replayed message. If ts is still valid, n_i will check the authenticity of *dREQ* through verifying $\hat{e}(DS,\, P) \overset{?}{=} \hat{e}(H_a(GS_k||ts||R),\, T)$ according to the following fact,

$$\hat{e}(DS, P) = \hat{e}(sH_a(GS_k||ts||R), P) = \hat{e}(H_a(GS_k||ts||R), sP)$$
$$= \hat{e}(H_a(GS_k||ts||R), T).$$

Here, $H_a(GS_k||ts||R)$ can be calculated using the information in *dREQ*. If *dREQ* passes the above verification, it proves that *dREQ* is authentic and n_i will accept *dREQ* and proceed to the data response phase. Otherwise, n_i will discard *dREQ*. Fig. 3 presents the data request process of *liteAGAP*.

C. Data Response Phase

First, each drone $n_i \in N$ retrieves its collected data d_i and calculates the hash value of d_i as $H_b(d_i)$. Then, n_i calculates its private key (the PUF response) through res_i = $rGen(m,\, che_i)$. After that, n_i creates its digital signature DS_i = res_iR + $res_iH_b(d_i)H_a(GS_k||ts||R)$, and sends a data response message $dREP_i$ = $\{d_i,\, DS_i\}$ to the aggregation drone $n_a \in N$. In this paper, we assume that a drone n_a is elected as the aggregation node according to the optimal cluster head selection strategy [35].

Second, after receiving *dREP*s from all other drones, n_a checks the authenticity of each $dREP_i$ ($n_i \in N$) through verifying $\hat{e}(DS_i,\, P) \overset{?}{=} \hat{e}(R,\, T_i)\hat{e}(H_a(GS_k||ts||R),\, T_i)^{H_b(d_i)}$ according to the following fact,

$$\begin{aligned}\hat{e}(DS_i, P) &= \hat{e}(res_iR + res_iH_b(d_i)H_a(GS_k||ts||R), P)\\ &= \hat{e}((R + H_b(d_i)H_a(GS_k||ts||R)), res_iP)\\ &= \hat{e}((R + H_b(d_i)H_a(GS_k||ts||R)), T_i)\\ &= \hat{e}((R, T_i)\hat{e}(H_a(GS_k||ts||R), T_i)^{H_b(d_i)}\end{aligned}$$

If $dREP_i$ passes the above verification, n_a will accept $dREP_i$. Otherwise, n_a will discard $dREP_i$ directly. After verifying all j - 1 data response messages, n_a adds its own data d_a and digital signature DS_a, and aggregates all j digital signatures into one aggregated digital signature DS^* = $\sum_{i=1}^{j} DS_i$. Finally, n_a sends all collected data D = $\{d_1,$ $d_2,\, \cdots,\, d_j\}$ and the aggregated digital signature DS^* to the ground station GS_k.

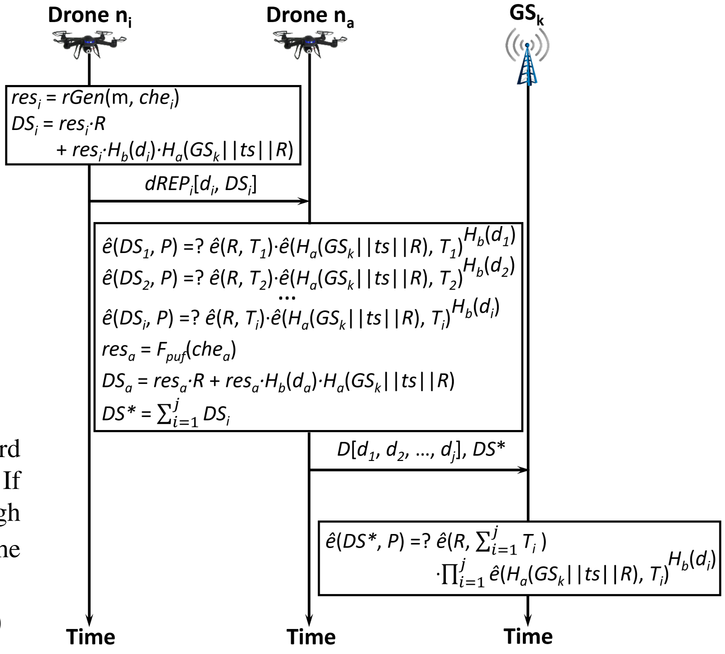

Drone n_i
Drone n_a
GS_k
res_i = rGen(m, che_i)
DS_i = res_i·R + res_i·H_b(d_i)·H_a(GS_k||ts||R)
dREP_i[d_i, DS_i]
ê(DS_1, P) =? ê(R, T_1)·ê(H_a(GS_k||ts||R), T_1)^H_b(d_1)
ê(DS_2, P) =? ê(R, T_2)·ê(H_a(GS_k||ts||R), T_2)^H_b(d_2)
...
ê(DS_i, P) =? ê(R, T_i)·ê(H_a(GS_k||ts||R), T_i)^H_b(d_i)
res_a = F_puf(che_a)
DS_a = res_a·R + res_a·H_b(d_a)·H_a(GS_k||ts||R)
DS* = Σ_{i=1}^j DS_i
D[d_1, d_2, ..., d_j], DS*
ê(DS*, P) =? ê(R, Σ_{i=1}^j T_i)
·Π_{i=1}^j ê(H_a(GS_k||ts||R), T_i)^H_b(d_i)
Time
Time
Time

Fig. 4. Data response phase.

Third, before the ground station GS_k accepts the collected data D, it has to verify the authenticity of aggregated digital signature DS^* by checking

$$\hat{e}(DS^*, P) \overset{?}{=} \hat{e}(R, \sum_{i=1}^{j} T_i) \prod_{i=1}^{j} \hat{e}(H_a(GS_k||ts||R), T_i)^{H_b(d_i)}$$

The correctness of above evaluation is based on the following

$$\begin{aligned}\hat{e}(DS^*, P) &= \hat{e}(\sum_{i=1}^{j} DS_i, P) = \prod_{i=1}^{j} \hat{e}(DS_i, P)\\ &= \prod_{i=1}^{j} \hat{e}((R + H_b(d_i)H_a(GS_k||ts||R)), T_i)\\ &= \hat{e}(R, \sum_{i=1}^{j} T_i) \prod_{i=1}^{j} \hat{e}(H_a(GS_k||ts||R), T_i)^{H_b(d_i)}\end{aligned}$$

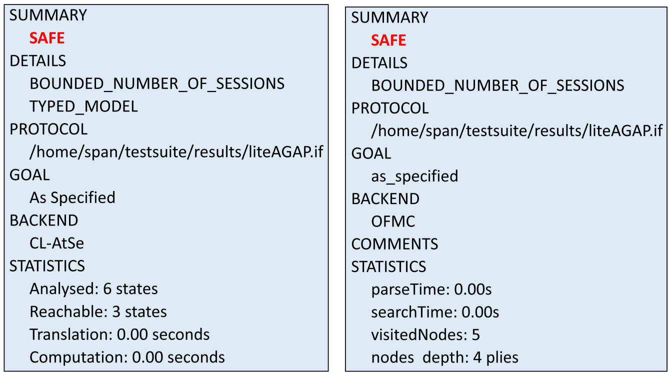
If DS^* passes the above verification, GS_k will accept D. Otherwise, GS_k will discards D directly. By this time, GS_k can use the collected data D for further processing and analysis. Fig. 4 presents the data response process of *liteAGAP*.

V. Security Verification and Informal Analysis

In this part, we first utilize AVISPA [16], which is an automated security scheme verification tool, to verify whether *liteAGAP* meets its security requirements and complies with all of AVISPA's security specifications. Through the automated security verification with AVISPA, we can demonstrate that no adversary can access or modify the critical information of *liteAGAP*. After that, we scrutinize the operations of *liteAGAP* in the context of different probable cyber attacks.

A. Security Verification Using AVISPA

In this subsection, we briefly explain the AVISPA tool, present the verification configuration and process, and show the verification results. AVISPA is an easy-to-use push-button



```
SUMMARY
    SAFE
DETAILS
    BOUNDED_NUMBER_OF_SESSIONS
    TYPED_MODEL
PROTOCOL
    /home/span/testsuite/results/liteAGAP.if
GOAL
    As Specified
BACKEND
    CL-AtSe
STATISTICS
    Analysed: 6 states
    Reachable: 3 states
    Translation: 0.00 seconds
    Computation: 0.00 seconds
```

(a)

```
SUMMARY
    SAFE
DETAILS
    BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
    /home/span/testsuite/results/liteAGAP.if
GOAL
    as_specified
BACKEND
    OFMC
COMMENTS
STATISTICS
    parseTime: 0.00s
    searchTime: 0.00s
    visitedNodes: 5
    nodes  depth: 4 plies
```

(b)

Fig. 5. Security verification results using AVISPA.

tool that uses the High-Level Protocol Specification Language (HLPSL) to realize internet security protocols. Here, HLPSL is a formal language that is used to model the communication behaviors of Internet security protocols. In addition, HLPSL can be used to clearly define the messages and specify their communication sequences, as well as outline the state transitions. AVISPA also offers multiple back-ends that employ automated analysis methods to check for the functional correctness of security protocol design.

First, we implement *liteAGAP* in HLPSL, and then select the CL-AtSe and OFMC back-ends [16] to test the security performance of *liteAGAP*. Here, CL-AtSe is a compositional logic back-end tool that is mainly used for threat modeling and vulnerability analysis. Speaking of OFMC, it is a back-end that checks for *liteAGAP*'s security properties such as confidentiality, authentication, and integrity. In the implemented HLPSL program, messages are exchanged between three entities, e.g., drone, ground station, and aggregation drone, and they are named as roles. Besides these three entity roles, some other roles, such as session, intruder, goal, and environment, are also defined in the HLPSL program. All roles in the HLPSL program will collectively help in checking for the security of *liteAGAP* via the CL-AtSe and OFMC back-ends. At the end, we run the HLPSL program in a virtual environment which is set up with Virtual Box. The Virtual Box is configured to run SPAN + AVISPA [36] on Debian Linux distribution Ubuntu 10.04. The verification results after running the HLPSL program using the CL-AtSe and OFMC back-ends are shown in Fig. 5. As observed, *liteAGAP* is safe from various security attacks, such as man-in-the-middle and replay attacks, and complies with all of AVISPA's security specifications. The HLPSL security verification programs are available at the https://github.com/congpu/liteAGAP.

B. Resilience and Immunity Analysis to Various Attacks

In this subsection, we informally demonstrate that *liteAGAP* is safe from various attacks such as drone impersonation attack, message modification attack, replay attack, drone capture attack, and ground station spoofing attack.

1) Drone Impersonation Attack: When an adversary pretends to be a legitimate drone n_i, it can fabricate a data response message *dREP* and send it to the aggregation drone. Even though the adversary is able to learn the identity of drone n_i through eavesdropping, it cannot obtain the valid PUF CRP of drone n_i because the PUF CRP is not piggybacked in the message and is securely stored by drone n_i and the ground station. Without the valid PUF CRP, the adversary will not be able to forge a legitimate digital signature that is required to generate a valid *dREP* message. Hence, *liteAGAP* is secure against drone impersonation attack.

2) Message Modification Attack: When an entity receives either data request message *dREQ* or data response message *dREP*, it first checks the validity of piggybacked sender's digital signature through $\hat{e}(DS, P) = \hat{e}(H_a(GSk||ts||R), T)$. If the validation succeeds, the receiving entity is certain that the message is valid and has not been modified maliciously. This ensures that *liteAGAP* is not vulnerable to message modification attack.

3) Drone Capture Attacks: Suppose that the adversary has successfully seized a legitimate drone n_i. Through power analysis attacks, the adversary is able to obtain drone n_i's identification and PUF challenge. The adversary also could attempt to retrieve the PUF response, however, this would be a wasted effort. This is because any change to the integrate circuit through power analysis attacks will inevitably change or even destroy the PUF, and the same response cannot be restored. Thus, the adversary is unable to attain drone n_i's PUF CRP, and cannot generate a valid digital signature which is recognized by the ground station. In addition, each drone has a unique PUF and the CRP values are also different, thus, capturing a single drone will not endanger other drones in the network.

4) Replay Attack: In the *liteAGAP*, the data request message *dREQ* is piggybacked with the current system time. When a drone n_i receives a *dREQ*, it first verifies whether the *dREQ* is obsolete. If the *dREQ* is fresh, it will continue to check for the authenticity of *dREQ*. Otherwise, it will discard the *dREQ* directly. Hence, *liteAGAP* is protected against replay attack.

5) Ground Station Spoofing Attack: Suppose that the adversary attempts to impersonate a ground station by sending a data request message *dREQ*. The adversary can generate a current timestamp and a random number, however, it cannot produce a valid digital signature which is required to sign of the *dREQ*. This is because the adversary does not have the private key of the ground station. The adversary can choose to fabricate a digital signature. Nevertheless, the fake digital signature can be easily detected by drones. Since the adversary cannot create a valid digital signature as well as drones will check for the validity of piggybacked digital signature in the *dREQ*, *liteAGAP* is immune to ground station spoofing attack.

VI. Performance Evaluation

A. Experimental Environment and Benchmarks

We conduct simulation-based experiments on an Apple M1 MacBook Air laptop with a memory of 8GB, running the Ventura macOS operating system. liteAGAP and other two benchmark schemes, PPAAS [17] and GASE [18], have been implemented in C programming language on Visual Studio

TABLE II
COMPARISON OF COMMUNICATION OVERHEAD*

Metrics	*liteAGAP*	**PPAAS**	**GASE**
Number of Msg.$^{\diamond}$	52	103	182
Size of Msg. (KB)‡	10.10	25.90	48.83
Energy Cons. (Joule)*	5.9×10^{-3}	11.6×10^{-3}	20.5×10^{-3}

*: We assume that there 50 entities (drone in *liteAGAP*, vehicles in PPAAS, and IoT nodes in GASE) in the network.

$^{\diamond}$: The total number of exchanged messages are obtained from the communication sequence diagrams provided by *liteAGAP*, PPAAS, and GASE.

‡: The total size of exchanged messages are calculated based on the implementation of *liteAGAP*, PPAAS, and GASE.

*: The energy communication of communication is computed with the total number of exchanged messages and the energy consumption of sending and receiving one message.

Code. We utilize the Pairing Based Cryptography (PBC) [37] library to perform mathematical operations required for bilinear pairing. A 256-bit hash function is chosen to generate the entity's digital signature; the hash function takes an input and maps to its corresponding group. The group defined for hash function H_a is $\mathbb{G}$, while the group for hash H_b is $\mathbb{Z}$.

For the performance comparison and analysis, we consider PPAAS [17] and GASE [18], and implement them in the same simulation environment. The fundamental ideas of PPAAS and GASE are discussed below.

1) PPAAS: PPAAS proposes an aggregation authentication scheme for the fog-to-cloud computing enabled vehicular ad hoc networks. PPAAS consists of five phases: a) initialization; b) registration; c) delivery; d) message processing; and e) trace. In the initialization phase, the trust authority generates the public parameters that will be used by the vehicles and the road-side units (RSUs). During the registration period, the vehicles and the RSUs register with the trust authority to get their secret information such as private-public key pair. In the delivery stage, the vehicles create their signcrypted messages and send them onto the RSUs. After that, the RSUs process the secret messages and store them along with the pseudonyms of vehicles. In the final phase, the RSUs trace and recover the real identity of malicious vehicles.

2) GASE: GASE aims to provide group authentication with session key agreement for edge computing devices. It has four stages: a) initialization; b) hashed secret sharing; c) group leader authentication; and d) server authentication. In the initialization stage, the secret dealer is selected, and the necessary environment is setup. Next, the nodes share one of their secrets within the group and are verified by the group leader. Finally, the group leader combines all the security tags and passes the aggregated tag onto the server who will perform aggregated tag verification. Note that GASE only provides group authentication for edge computing devices. On the other hand, our approach *liteAGAP* not only provides authentication, but also allows drones to submit the collected data to the ground station.

We evaluate the performance of *liteAGAP*, PPAAS, and GASE, and obtain the results of communication overhead, running time, memory space usage, and energy consumption by varying the number of executed algorithms and the number of drones. Communication overhead is represented through the number of exchanged messages, the size of exchanged messages, as well as the energy consumption of exchanged messages. Running time represents the time elapsed from when the protocol starts running to when it stops running. Memory space usage (or RAM usage) is amount of memory required to run the protocol. Energy consumption indicates how much energy is consumed by the protocol.

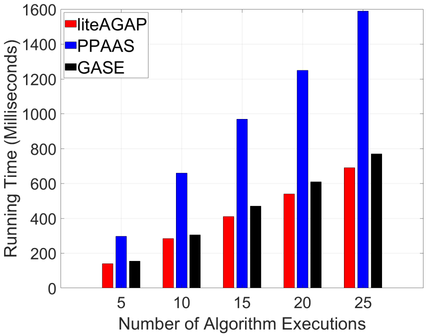


Fig. 6. Running time versus the number of algorithm executions.

B. Experimental Results and Analysis

First, we obtain the communication overhead of *liteAGAP*, PPAAS, and GASE in terms of the total number of exchanged messages, the total size of exchanged messages, as well as the energy consumption of communication, and then present the results in Table II. In this experiment, we assume that there are 50 entities (i.e., drones in *liteAGAP*, vehicles in PPAAS, and IoT nodes in GASE) in the network. For the total number of exchanged messages, we investigate the communication sequence diagrams in *liteAGAP*, PPAAS, and GASE, and directly count how many messages are needed for each security protocol. In our approach *liteAGAP*, one data request message is first sent by the ground station. After receiving the data request message, each drone in the group replies one data response message. Since we consider 50 drones in the group, 50 data response messages are generated and sent. Then, the aggregation drone sends one aggregated message to the ground station. In summary, 52 messages are needed by our approach *liteAGAP*. In PPAAS, the trust authority first sends a message piggybacked with the pseudonym-partial private key pair to each vehicle in the network (50 messages are needed for 50 vehicles). Then, each vehicle can use the private key pair to encrypt traffic related message and send it to the road-side unit (50 messages are sent by 50 vehicles). Finally, one message will be needed for the broadcast of safety warning message, aggregated message, and pseudonym of malicious vehicles, respectively. Thus, 103 messages are required to complete the entire process of PPAAS. GASE requires the largest number of messages to be exchanged between the communication entities in the network, where a total of 182 messages are sent. According to the real implementation in C programming language, the total size of exchanged messages is 10.10 KB, 25.90 KB, and 48.83 KB for *liteAGAP*, PPAAS, and GASE, respectively. We also calculate

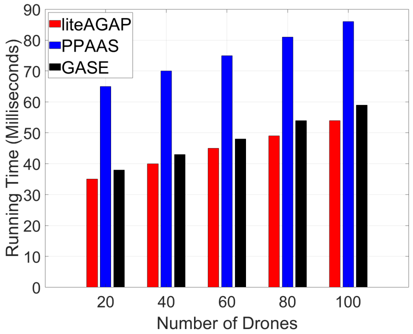


Fig. 7. Running time versus the number of drones.

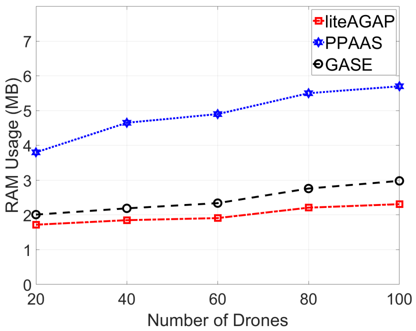


Fig. 9. RAM usage versus the number of drones.

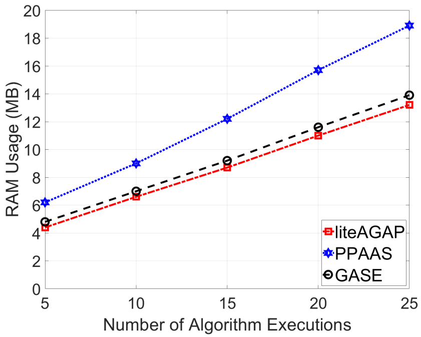


Fig. 8. RAM usage versus the number of algorithm executions.

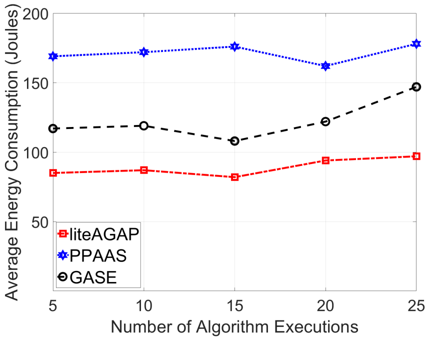


Fig. 10. Average energy consumption versus the number of algorithm executions.

the energy consumption of communication based on the total number of exchanged messages and the energy consumption of sending and receiving one message [38]. It is relatively straightforward that our approach *liteAGAP* consumes the least amount of energy compared to PPAAS and GASE. This is because the least number of messages are sent by 50 drones in our approach *liteAGAP*.

Second, we measure the running time of *liteAGAP*, PPAAS, and GASE by changing the number of algorithm executions in Fig. 6. Overall, the running time of *liteAGAP*, PPAAS, and GASE increase as the number of algorithm executions is increased from 5 to 25. This is because when the security protocol is executed more times, a longer running time is expected. In addition, it is clearly shown that the longest running time belongs to PPAAS. Since a road-side unit needs to decrypt each signcrypted traffic-related message and then aggregate them, a longer running time is required by PPAAS. Our approach *liteAGAP* outperforms PPAAS and GASE because lightweight cryptographic operations such as bilinear pairing, hash function, and physical unclonable function are adopted. Moreover, the ground station in our approach *liteAGAP* does not decrypt the secret messages from drones. A longer running time is observed by GASE than that of *liteAGAP* because of a large number of hashed-secrets sharing. The running time is also measured with varying number of drones in Fig. 7. Generally, as the number of drones in the network is increased from 20 to 100, the running time of all three schemes increase linearly. Since a larger number of drones will produce more data packets in the network, thus, a longer running time is required for authentication and aggregation operations. However, our approach *liteAGAP* still provides the best performance compared to PPAAS and GASE because of the adoption of execution-efficient operations.

Third, we measure the RAM usage of all three schemes by changing the number of algorithm executions and the number of drones, and present the results in Fig. 8 and 9, respectively. Here, the RAM usage indicate how much memory space is utilized to run the security protocol. As observed in Fig. 8 and 9, our approach *liteAGAP* consumes the least amount of memory space, while the largest amount of memory space is consumed by PPAAS. The rationale is that PPAAS requires the road-side units to decrypt each encrypted message before aggregation, and more data packets need to be stored temporarily. As a result, more memory space is required by PPAAS. Our approach *liteAGAP* shows the lowest RAM usage because the aggregation drone aggregates the encrypted data directly, and then sends them to the ground station. Thus, it consumes the least amount of memory space.

Finally, we measure the average energy consumption of *liteAGAP*, PPAAS, and GASE against the number of algorithm executions in Fig. 10. Overall, our approach *liteAGAP* provides a lower energy consumption compared to PPAAS and GASE. *liteAGAP* makes use of lightweight operations such as bilinear pairing, physical unclonable function, and hash function, and directly aggregates the encrypted messages from drones without decryption. Thus, the least amount of energy is consumed by our approach *liteAGAP*. GASE consumes more energy than our approach *liteAGAP* because it requires a large number of IoT nodes to compute their tokens with the random numbers and shadow secrets. PPAAS delivers the largest amount of energy consumption. This is because the road-side units frequently decrypt the messages from vehicles in the network before the aggregation operations.

VII. Conclusion

In this paper, a lightweight aggregate authentication scheme (*liteAGAP*) is designed for IoD systems, where a group of drone send their data along with their digital signatures to an aggregation drone. After that, the aggregation drone combines the received digital signatures along with the data, and then send them to the ground station. *liteAGAP* not only improves the security of IoD data communication, but also realizes the IoD data exchange in a more efficient way. In addition, considering the constrained-recourse of drones, we chose cryptographic primitives such as physical unclonable function, bilinear pairing, and hash function to realize *liteAGAP*. In order to prove that *liteAGAP* is safe from cyber attacks and free of security design vulnerabilities, we not only conducted a systematic security verification using AVISPA, but also performed a security analysis informally. Furthermore, we conducted a comprehensive simulation-based experimental study to evaluate the performance of *liteAGAP*, and compared it with other two benchmark schemes. The experimental findings show that our approach *liteAGAP* outperforms the existing schemes, indicating a practical solution for IoD applications.

References

[1] C. Pu and K. Choo, "Lightweight Sybil Attack Detection in IoT based on Bloom Filter and Physical Unclonable Function," *Elsevier Computers & Security*, vol. 113, p. 102541, 2022.

[2] C. Pu, I. Ahmed, E. Allen, and K. Choo, "A Stochastic Packet Forwarding Algorithm in Flying Ad Hoc Networks: Design, Analysis, and Evaluation," *IEEE Access*, vol. 9, pp. 162 614–162 632, 2021.

[3] S. Krishnan and M. Murugappan, *Internet of Drones: Applications, Opportunities, and Challenges*. CRC Press, 2023.

[4] C. Pu, "A Reinforcement Learning Based Service Scheduling Algorithm for Internet of Drones," in *Proc. IEEE ICC Workshops*, 2022, pp. 999–1004.

[5] C. Pu and L. Carpenter, "To Route or To Ferry: A Hybrid Packet Forwarding Algorithm in Flying Ad Hoc Networks," in *Proc. IEEE NCA*, 2019, pp. 1–8.

[6] C. Pu, "A novel blockchain-based trust management scheme for vehicular networks," in *Proc. IEEE WTS*, 2021, pp. 1–6.

[7] M. Pathak, D. Dwivedi, N. Kaur, V. Chaturvedi, A. Dwivedi, R. Singh, R. Kumar, M. Patel, and H. Sharan, *Application of Cognitive Internet of Things (IoT) for COVID-19 Pandemic*. Chapman and Hall/CRC, 2022.

[8] C. Pu and L. Carpenter, "Psched: A Priority-Based Service Scheduling Scheme for the Internet of Drones," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4230–4239, 2020.

[9] C. Pu, A. Wall, K. Choo, I. Ahmed, and S. Lim, "A Lightweight and Privacy-Preserving Mutual Authentication and Key Agreement Protocol for Internet of Drones Environment," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9918–9933, 2022.

[10] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-Air Computing for Wireless Data Aggregation in Massive IoT," *IEEE Wireless Communications*, vol. 28, no. 4, pp. 57–65, 2021.

[11] J. Wang, L. Wu, S. Zeadally, M. Khan, and D. He, "Privacy-preserving Data Aggregation against Malicious Data Mining Attack for IoT-enabled Smart Grid," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–25, 2021.

[12] A. Ullah, M. Azeem, H. Ashraf, A. Alaboudi, M. Humayun, and N. Jhanjhi, "Secure Healthcare Data Aggregation and Transmission in IoT—A Survey," *IEEE Access*, vol. 9, pp. 16 849–16 865, 2021.

[13] A. Berini, M. Ferrag, B. Farou, and H. Seridi, "HCALA: Hyperelliptic curve-based anonymous lightweight authentication scheme for Internet of Drones," *Pervasive and Mobile Computing*, vol. 92, p. 101798, 2023.

[14] D. Chaudhary, T. Soni, K. Vasudev, and K. Saleem, "A modified lightweight authenticated key agreement protocol for Internet of Drones," *Internet of Things*, vol. 21, p. 100669, 2023.

[15] C. Pu, A. Wall, I. Ahmed, and K. Choo, "SecureIoD: A Secure Data Collection and Storage Mechanism for Internet of Drones," in *Proc. IEEE MDM*, 2022, pp. 83–92.

[16] *Automated Validation of Internet Security Protocols and Applications*, Last accessed: Jan 26, 2022, http://www.avispa-project.org/.

[17] Y. Yang, L. Zhang, Y. Zhao, K. Choo, and Y. Zhang, "Privacy-Preserving Aggregation-Authentication Scheme for Safety Warning System in Fog-Cloud Based VANET," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 317–331, 2022.

[18] M. Nakkar, R. AlTawy, and A. Youssef, "GASE: A Lightweight Group Authentication Scheme With Key Agreement for Edge Computing Applications," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 840–854, 2023.

[19] *Internet-of-Drones: Novel Applications, Recent Deployments and Integration*, Last accessed: June 10, 2023, https://www.comsoc.org/publications/magazines/ieee-internet-things-magazine/cfp/internet-drones-novel-applications-recent.

[20] *The sky is not the limit: The past, present, and future of the Internet of Drones*, Last accessed: June 10, 2023, https://www.ericsson.com/en/blog/2021/6/internet-of-drones-sky-is-not-the-limit.

[21] G. Bansal and B. Sikdar, "S-MAPS: Scalable Mutual Authentication Protocol for Dynamic UAV Swarms," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12 088–12 100, 2021.

[22] S. Hussain, S. Chaudhry, O. Alomari, M. Alsharif, M. Khan, and N. Kumar, "Amassing the Security: An ECC-Based Authentication Scheme for Internet of Drones," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4431–4438, 2021.

[23] A. Yazdinejadna, R. Parizi, A. Dehghantanha, and H. Karimipour, "Federated learning for drone authentication," *Ad Hoc Networks*, vol. 120, p. 102574, 2021.

[24] Y. Aydin, G. Kurt, E. Ozdemir, and H. Yanikomeroglu, "Group Authentication for Drone Swarms," in *Proc. IEEE WiSEE*, 2021, pp. 72–77.

[25] M. Abdel-Malek, K. Akkaya, A. Bhuyan, and A. Ibrahim, "A Proxy Signature-Based Drone Authentication in 5G D2D Networks," in *Proc. IEEE VTC2021-Spring*, 2021, pp. 1–7.

[26] C. Lai and Z. Chen, "Group-based Handover Authentication for Space-Air-Ground Integrated Vehicular Networks," in *Proc. IEEE ICC*, 2021, pp. 1–6.

[27] Y. Tan, J. Wang, J. Liu, and N. Kato, "Blockchain-Assisted Distributed and Lightweight Authentication Service for Industrial Unmanned Aerial Vehicles," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 16 928–16 940, 2022.

[28] M. El-Zawawy, A. Brighente, and M. Conti, "SETCAP: Service-Based Energy-Efficient Temporal Credential Authentication Protocol for Internet of Drones," *Computer Networks*, vol. 206, p. 108804, 2022.

[29] *How Police Drones Technology Can Be Used at a Protest*, Last accessed: Sep 15, 2022, https://privacyinternational.org/explainer/4498/how-police-drones-technology-can-be-used-protest.

[30] Q. Do, B. Martini, and K. Choo, "The role of the adversary model in applied security research," *Computers & Security*, vol. 81, pp. 156–181, 2019.

[31] C. Pu, A. Wall, and K. Choo, "Bilinear Pairing and PUF Based Lightweight Authentication Protocol for IoD Environment," in *Proc. IEEE MASS*, 2022, pp. 115–121.

[32] C. Pu, H. Zerkle, A. Wall, S. Lim, K. Choo, and I. Ahmed, "A Lightweight and Anonymous Authentication and Key Agreement Protocol for Wireless Body Area Networks," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21 136–21 146, 2022.

[33] J. Wallrabenstein, "Practical and Secure IoT Device Authentication Using Physical Unclonable Functions," in *Proc. IEEE FiCloud*, 2016, pp. 99–106.

[34] F. MacWilliams and N. Sloane, *The Theory of Error Correcting Codes*. Elsevier, 1977.

[35] D. Zhang, H. Ge, T. Zhang, Y. Cui, X. Liu, and G. Mao, "New Multi-hop Clustering Algorithm for Vehicular Ad Hoc Networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1517–1530, 2018.

[36] *SPAN*, Last accessed: April 25, 2022, http://people.irisa.fr/Thomas.Genet/span/.

[37] *PBCLibrary*, https://crypto.stanford.edu/pbc/.

[38] C. Pu and S. Lim, "A Light-Weight Countermeasure to Forwarding Misbehavior in Wireless Sensor Networks: Design, Analysis, and Evaluation," *IEEE Systems Journal*, vol. 12, no. 1, pp. 834–842, 2018.